

Een “volwassen” REST-voorbeeld toegepast op StUF

Auteur: Henri Korver

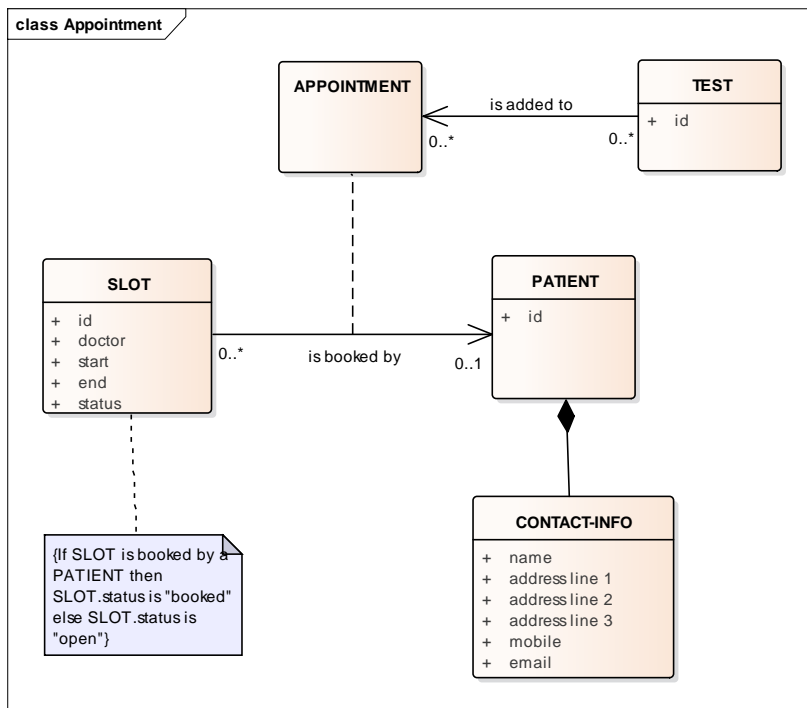
Inleiding

In zijn artikel [“Richardson Maturity Model: steps towards the glory of REST”](http://martinfowler.com/articles/richardsonMaturityModel.html) legt Martin Fowler drie volwassenheidsniveaus van REST-toepassingen uit aan de hand van een (fictief) voorbeeld waarin REST-services worden gebruikt om een afspraak te maken bij de dokter.¹ In deze notitie vertalen we Fowlers berichten die horen bij het hoogste niveau (Level 3 - Hypermedia Controls) naar de context van StUF. Dit niveau wordt vaak aangeduid met het acroniem HATEOAS (Hypertext As The Engine Of Application State). In het voorbeeld van Fowler beantwoordt het de vraag wat je moet doen om beginnend met een lijst van beschikbare tijden een afspraak te maken met een dokter met eventuele vervolgstappen, zoals het afzeggen of wijzigen van de afspraak, of het toevoegen van testresultaten.

Semantisch model

StUF gaat ervan uit dat de berichten worden gedefinieerd op basis van een semantisch model. In Figuur 1 heb ik het model proberen te reconstrueren dat Fowler (ongeveer) in zijn hoofd moet hebben gehad toen hij zijn REST-berichten aan het opstellen was.

¹ <http://martinfowler.com/articles/richardsonMaturityModel.html>



Figuur 1: UML model voor een afspraak met de dokter

Een patiënt (PATIENT) kan een open tijdvak (SLOT) boeken in de agenda van een dokter. Als dat is gebeurd wordt de status van het tijdvak omgezet van “open” naar “booked”. Op het moment dat een tijdvak is gereserveerd door een patiënt spreken we over een afspraak (APPOINTMENT). Aan een afspraak kunnen diverse tests (bloedonderzoek, etc.) worden gekoppeld.

De onderstaande tabel geeft aan hoe de StUF-entiteitstypen vertaald worden naar de URI's uit de voorbeelden van Fowler.

Entiteitstype	URI
DOCTOR	/docs
PATIENT	/patients
SLOT	/slots
APPOINTMENT	/slots/{id}/appointment

Tabel 1

De mapping van appointment is een speciaal geval omdat het een relatie-entiteit (in UML: association class). Als je het gereserveerde tijdvak (slot) kent door middel van de {id} dan weet je ook de bijbehorende afspraak.

Raadplegen beschikbaarheid

Met de volgende GET-request wordt de beschikbaarheid van dokter “mjones” opgevraagd.

GET /doctors/mjones/slots?date=20100104&status=open HTTP/1.1
Host: royalhope.nhs.uk

Opmerking [HK1]: To do: Neem ook het echte StUF-vraagbericht op waarnaar dit vertaald moet worden.

In StUF 3.02 zouden we deze GET-request integraal kunnen overnemen.

In onderstaand responsbericht van Fowler wordt door middel van open slots de beschikbaarheid van dokter mjones gerepresenteerd.

HTTP/1.1 200 OK
[various headers]

```
<openSlotList>
  <slot id="1234" doctor="mjones" start="1400" end="1450">
    <link rel="/linkrels/slot/book" uri="/slots/1234"/>
  </slot>
  <slot id="5678" doctor="mjones" start="1600" end="1650">
    <link rel="/linkrels/slot/book" uri="/slots/5678"/>
  </slot>
</openSlotList>
```

In (de nieuwe) StUF zou dit bericht er als volgt uit kunnen zien:²

HTTP/1.1 200 OK
[various headers]

```
<openSlotList functie="lijst">
  <slot entiteittype="SLOT" functie="entiteit">
    <id>1234</id>
    <doctor>mjones</doctor>
    <begin>1400</begin>
    <end>1450</end>
    <link rel="/linkrels/slot/book" uri="/slots/1234"/>
  </slot>
  <slot entiteittype="SLOT" functie="entiteit">
    <id>5678</id>
    <doctor>mjones</doctor>
    <begin>16</begin>
    <end>1650</end>
    <link rel="/linkrels/slot/book" uri="/slots/5678"/>
  </slot>
</openSlotList>
```

We zien de volgende verschillen tussen de twee bovenstaande berichten:

- In het StUF-bericht worden de gereserveerde attributen zoals “entiteittype” en “functie” gebruikt om de verbinding te leggen met het semantisch model zoals weergegeven in Figuur 1.³
- In het StUF-bericht worden de eigenschappen van het entiteit SLOT opgenomen als elementen en niet als attributen. In StUF zijn attributen altijd gereserveerd en hebben ze van te voren gedefinieerde functionaliteit.

Om het bovenstaande StUF bericht mogelijk te maken moeten er nog twee RFC's worden ingediend:

² Voor de leesbaarheid zijn de namespace-declaraties en -prefixes weggelaten.

³ Het attribuut functie="entiteit" is default en mag worden weggelaten in bovenstaand StUF-bericht.

- Het opnemen van nul of meer `<link rel="..." uri="...">` elementen binnen een entiteit-element.
- Uitbreiding van het waardenbereik van het attribuut functie met de waarde "lijst". Hiermee kan worden aangegeven dat het element een container is voor een lijst met entiteiten van het type zoals aangegeven in het attribuut entiteittype. Het voorstel is om in [RFC0129](#) (Uitbreiding waardenbereik attribuut "StUF:functie") de waarde "lijst" toe te voegen.

Boeken van een afspraak

Voor het boeken van het open slot "1234" bij dokter mjones gebruikt Fowler het volgende bericht:

POST /slots/1234 HTTP/1.1
[various other headers]

```
<appointmentRequest>
  <patient id="jsmith"/>
</appointmentRequest>
```

In StUF zou dit bericht er als volgt uitzien:

POST /slots/1234 HTTP/1.1
[various other headers]

```
<appointmentRequest functie="container">
  <patient entiteittype="PATIENT">
    <id>jsmith</id>
  </patient>
</appointmentRequest>
```

In het StUF-bericht is het element "appointmentRequest" een betekenisloze container om meerdere elementen op te kunnen nemen. In dit geval zou het container-element weggelaten kunnen worden omdat het toch maar één element bevat. Hetzelfde geldt voor het bericht van Fowler. In het element "patient" van het StUF-bericht hebben we het attribuut functie="entiteit" weggelaten omdat het default is.

Het responsbericht van Fowler ziet er zo uit:

HTTP/1.1 201 Created
Location: http://royalhope.nhs.uk/slots/1234/appointment
[various headers]

```
<appointment>
  <slot id="1234" doctor="mjones" start="1400" end="1450"/>
  <patient id="jsmith"/>
  <link rel="linkrels/appointment/cancel" uri="/slots/1234/appointment"/>
  <link rel="linkrels/appointment/addTest" uri="/slots/1234/appointment/tests"/>
  <link rel="self" uri="/slots/1234/appointment"/>
  <link rel="linkrels/appointment/changeTime"
    uri="/doctors/mjones/slots?date=20100104@status=open"/>
  <link rel="linkrels/appointment/updateContactInfo" uri="/patients/jsmith/contactInfo"/>
  <link rel="linkrels/help" uri="/help/appointment"/>
</appointment>
```

In StUF ziet de respons er zo uit:

HTTP/1.1 201 Created
Location: http://royalhope.nhs.uk/slots/1234/appointment
[various headers]

```

<appointment entiteittype="APPOINTMENT" functie="relatie-entiteit">
  <slot entiteittype="SLOT">
    <id>1234</id>
    <doctor>mjones</doctor>
    <begin>1400</begin>
    <end>1450</end>
  </slot>
  <patient entiteittype="PATIENT">
    <id>jsmith</id>
  </patient>
  <link rel="/linkrels/appointment/cancel" uri="/slots/1234/appointment"/>
  <link rel="/linkrels/appointment/addTest" uri="/slots/1234/appointment/tests"/>
  <link rel="self" uri="/slots/1234/appointment"/>
  <link rel="/linkrels/appointment/changeTime"
    uri="/doctors/mjones/slots?date=20100104@status=open"/>
  <link rel="/linkrels/appointment/updateContactInfo" uri="/patients/jsmith/contactInfo"/>
  <link rel="/linkrels/help" uri="/help/appointment"/>
</appointment>

```

In eerdere berichten zagen we al “lijst” en “entiteit” voorkomen als waarden van het attribuut functie. In bovenstaand voorbeeld is de nieuwe waarde “relatie-entiteit” daaraan toegevoegd. Hiermee kan worden aangegeven dat APPOINTMENT fungeert als relatie-entiteit c.q. association class tussen SLOT en PATIENT zoals weergegeven in het UML-model van Figuur 1. Het toevoegen van waarde “relatie-entiteit” is een aanvulling op [RFC0129](#) (Uitbreiding waardenbereik attribuut "StUF:functie").

To do

- De echte RFC is dat we met container elementen ook de echte vrije berichten kunnen introduceren, dus helemaal zonder stuurgegevens.
- Functie “gerelateerde” voor impliciete relaties is in bovenstaand verhaal nog niet aanbod gekomen maar deze kan ook heel waardevol zijn voor het opstellen van compacte berichten à la REST en moet wellicht ook verder uitgewerkt worden in een ander voorbeeld en als RFC hersteld worden.